Take-home question

(50 pts) 1. Wikipedia defines a set as follows:

In computer science, a set is an abstract data type that can store certain values, without any particular order, and no repeated values. It is a computer implementation of the mathematical concept of a finite set. Unlike most other collection types, rather than retrieving a specific element from a set, one typically tests a value for membership in a set.

In a set there isn’t a meaningful distinction between key and value (they are the same). An item of type T can be inserted into the set and removed from the set, and duplicate insertions are ignored. You can find examples in C++ in the set library:

A set typically has the following operations:
- insert - insert an item in the set
- delete - remove an item from the set
- find - tests to see if an item is in the set

For the take-home portion of your exam (worth half of the final’s points value and 5% of your final grade) I would like you to prepare a report comparing several data structure implementations of a set ADT. From class this semester, our common culprits are:

- as a dynamic array
- as a linked list
- as a binary tree
- as a hash table
- any other structures covered that would serve as appropriate data structure representations

In your report I would like you to address:

- What the implementation would look like. Please provide examples (descriptions, diagrams if necessary) to clarify what your implementation would look like internally.
- How would the elementary operations be performed (with Big-O runtime of operations)? What would the resulting Big-O be for each?
• Finally, rank each implementation according to applicability. Justify your ranking.

The sort of analysis I am looking for is precisely what we have been doing all semester when we encounter a new data structure or ADT.

*Despite my lengthy description, a reasonable target for this report would be no more than 1-2 pages. Be specific, and terse. I will be assigning credit based on appropriateness of your analysis, not word count.*